

Министерство науки и высшего образования Российской Федерации
 Федеральное государственное автономное образовательное учреждение высшего образования
 «Новосибирский национальный исследовательский государственный университет»
 (Новосибирский государственный университет, НГУ)

Физический факультет

Согласовано, декан ФФ
 Блиннов В.Е.
 подпись
 «07» 04 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Основы программирования

Направление подготовки: **03.03.02 Физика**

Направленность (профиль): **Физика. Фундаментальная и экспериментальная физика**

Направление подготовки **03.03.01 Прикладные математика и физика**

Направленность (профиль): **Прикладные математика и физика. Прикладная физика и инженерия**

Форма обучения: очная

Семестр	Общий объем	Виды учебных занятий (в часах)					Промежуточная аттестация (в часах)					
		Контактная работа обучающихся с преподавателем				Самостоятельная работа, не включая период сессии	Самостоятельная подготовка к промежуточной аттестации	Контактная работа обучающихся с преподавателем				
		Лекции	Практические занятия	Лабораторные занятия	Индивидуальная работа с преподавателем/ Консультации в период занятий			Консультации	Зачет	Дифференцированный зачет	Экзамен	
1	2	3	4	6	7	8	9	10	11	12	13	
2	108	16	48			42					2	
Всего 108 часов /3 зачетные единицы, из них: контактная работа 66 часов												
Компетенции: ОПК-3 (СУОС 03.03.02 Физика) ОПК-4 (ФГОС 03.03.01 Прикладные математика и физика)												

Руководитель программы,
 д.ф.-м.н., проф



С.В.Цыбуля

Новосибирск, 2026

Содержание

1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с установленными в программе индикаторами достижения компетенций	3
2. Место дисциплины в структуре образовательной программы	3
3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося	4
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий	4
5. Перечень учебной литературы	6
6. Перечень учебно-методических материалов по самостоятельной работе обучающихся	7
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	7
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.....	7
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.....	7
10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине	8
Приложение 1_ОТЧЕТ О РАБОТЕ	Ошибка! Залка не определена.

1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с установленными в программе индикаторами достижения компетенций

При освоении дисциплины «**Основы программирования**» студенты получают базовые знания о принципах программирования и обучение программному конструированию на языке программирования С.

Дисциплина нацелена на формирование у обучающегося общепрофессиональной компетенции

Результаты освоения образовательной программы (компетенции)	Индикаторы	Результаты обучения по дисциплине
<p>ОПК-3. (СУОС 03.03.02 Физика) Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.</p>	<p>ОПК – 3.2. Применяет основные приемы, возможности и правила работы со стандартными и специализированными программными продуктами при решении профессиональных задач.</p>	<p>Знать: фундаментальные основы структурного подхода в программировании; основы работы с файловым вводом-выводом; принципы разработки консольных приложений; основы текстовой графики; принципы организации современных вычислительных систем.</p>
<p>ОПК-4. (ФГОС 03.03.01 Прикладные математика и физика) Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач</p>	<p>ОПК - 4.2. Применяет программно-технологические платформы, сервисы и информационные ресурсы информационной системы, в том числе, специализированных патентных базах данных для обработки научной и технической информации при решении фундаментальных и прикладных задач.</p>	<p>Уметь оперировать с основными типами данных и конструировать из них новые типы; работать с динамическими структурами данных: массивами и списками; оценивать сложность алгоритмов по времени и по памяти; ставить корректные технические задания. Владеть принципами структурного и объектно-базированного программирования; возможностью оценивать трудозатраты по программной реализации задачи.</p>

2. Место дисциплины в структуре образовательной программы

В результате прохождения курса студенты должны овладеть принципами структурного программирования, а также научиться оценивать трудозатраты по программной реализации задачи.

Для успешного освоения курса необходимо знание английского языка на уровне чтения текста и основ линейной алгебры.

После изучения курса студенты могут продолжить изучать программирование на курсах, посвященных объектно-ориентированным языкам или практическому программированию в физических приложениях.

3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Трудоемкость дисциплины составляет: 3 зачетные единицы, 108 часов.

Форма промежуточной аттестации: 2 семестр – дифференцированный зачет

№	Вид деятельности	Семестр
		2
1	Лекции, ч	16
2	Практические занятия, ч	48
3	Лабораторные занятия, ч	
4	Занятия в контактной форме, ч из них	66
5	из них аудиторных занятий, ч	64
6	в электронной форме, ч	-
7	консультаций, час.	
8	промежуточная аттестация, ч	2
9	Самостоятельная работа, час.	42
10	Всего, ч	108

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Лекции (16 ч)

Наименование темы и их содержание	Объем, час
Лекция 1. <i>Основы структурного программирования. Подход к решению задач.</i> Задачи и области применения программирования и языка С. Основные управляющие конструкции алгоритмических языков (оператор, последовательность, переменная, ветвление, цикл, вложенные циклы).	2
Лекция 2. <i>Функции и массивы данных. Динамические массивы. Указатели</i> Создание пользовательских функций. Локальные и глобальные переменные. Возвращение значений функциями. Передача параметров в функции.	2
Лекция 3. <i>Динамические массивы. Указатели</i> Выделение и освобождение памяти. Одномерные и многомерные массивы данных. Арифметика указателей. Особенности работы с массивами через арифметику указателей.	2

<p>Лекция 4. Указатели и ссылки. Алгоритм сортировки пузырьком Ссылки. Передача параметров по ссылке. Константы. Алгоритмы сортировки. Алгоритм сортировки пузырьком и его оптимизации. Быстрые алгоритмы сортировки. Математическая сложность алгоритмов. Алгоритмы, работающие с предварительной информацией о массивах.</p>	2
<p>Лекция 5. Системы счисления данных. Позиционные и непозиционные системы счисления. Исторический экскурс. Двоичная и шестнадцатеричная системы как основа в информатике. Алгоритмы перевода между системами счисления.</p>	2
<p>Лекция 6. Пользовательские типы данных Символьный тип данных. Строки в языке Си как одномерный массив символов. Кодировки. Структурный тип данных. Объединение разнородных данных как основа структурного подхода в программировании.</p>	2
<p>Лекция 7. Списки структур данных Способы оперирования со структурными типами данных. Односвязные и двусвязные списки данных. Кольцевой список и кольцевой список с барьером. Деревья. Бинарные деревья. Добавление и удаление элементов в бинарных деревьях. Поиск по бинарным деревьям.</p>	2
<p>Лекция 8. Работа с файлами. Контроль динамической памяти Понятие файла и файловой системы. Текстовые, бинарные и исполняемые файлы. Файл как поток. Буферизация данных при записи в файл. MemoryLeaks как критичная ошибка для программ, предназначенных для научных экспериментов. Способы контроля утечек памяти.</p>	2

Практические занятия (48 ч)

Содержание практического занятия	Объем, час
<p>Практическое занятие 1. Задачи и области применения программирования и языка С. Основы языка С. Структура программы. Типы данных. Ввод-вывод. Условный оператор IF. Цикл FOR. Форматирование кода.</p>	3
<p>Практическое занятие 2. Системы счисления. Бинарные операции. Хранение данных в ПК.</p>	3
<p>Практическое занятие 3. Точность и погрешность. Типы данных и их предельные значения (int, unsigned int, float, double).</p>	3
<p>Практическое занятие 4. Форматированный вывод и работа с printf(). Округление и вывод заданного количества знаков.</p>	3

Практическое занятие 5. Структурный подход в программировании. Функции. Глобальные и локальные переменные. Передача параметров.	3
Практическое занятие 6. Одномерные и многомерные массивы данных. Препроцессор и директива #define.	3
Практическое занятие 7. Сортировка введенного динамического массива, реализация алгоритма сортировки пузырьком и его оптимизации.	3
Практическое занятие 8. Реализация единой функции сортировки в соответствии с математической моделью, за счет выделения функций IfRepace() и Replace().	3
Практическое занятие 9. Обсуждение особенностей выполнения заданий. Сравнение различных реализаций. Выявление достоинств и недостатков каждой реализации.	3
Практическое занятие 10. Символьный тип данных и строки. Таблица ASCII.	3
Практическое занятие 11. Случайные и псевдослучайные числа. Работа с генератором случайных чисел в С.	3
Практическое занятие 12. Обработка текста. Алгоритмика применения правил русского языка на примере текста, введенного пользователем.	3
Практическое занятие 13. Вычисление определителя с матриц помощью рекурсивных функций.	3
Практическое занятие 14. Вычисление обратной матрицы, как транспонированной матрицы алгебраических дополнений.	3
Практическое занятие 15. Методы контроля утечек памяти в программах.	3
Практическое занятие 16. Способы оценки трудоемкости программ. Стандартные методы определения узких и сложных мест в программах. Профилировщики кода.	3

Самостоятельная работа студентов (42ч)

Перечень занятий на СРС	Объем, час
Подготовка к практическим занятиям.	22
Изучение разделов дисциплины по учебной литературе, в том числе вопросов, не освещаемых на лекциях.	20

5. Перечень учебной литературы

1. Двойнишников С.В., Лысаков К. Ф. Основы программирования (язык С): Учеб. пособие / Новосиб. гос. ун-т. Новосибирск: ИПЦ НГУ, 2018. – 158 с. (40 экз.)
2. Вирт, Никлаус Алгоритмы и структуры данных: с примерами на Паскале / Никлаус Вирт ; [пер. с англ. Д. Б. Подшивалова 2-е изд., испр.] Санкт-Петербург : Невский Диалект, 2007 350, [1] с. : ил. ; 20 см. (Библиотека программиста) ISBN 5-7940-0065-1. (26 экз.)

6. Перечень учебно-методических материалов по самостоятельной работе обучающихся

3. Д. Э. Кнут: Искусство Программирования. Том 1 Основные Алгоритмы, 1976
4. Д. Э. Кнут: Искусство Программирования. Том 2 Получисленные алгоритмы, 1976
5. Д. Э. Кнут: Искусство Программирования. Том 3 Сортировка и Поиск, 1976

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

7.1 Ресурсы сети Интернет

Для освоения дисциплины используются следующие ресурсы:

- электронная информационно-образовательная среда НГУ (ЭИОС);
- образовательные интернет-порталы;
- информационно-телекоммуникационная сеть Интернет.

Взаимодействие обучающегося с преподавателем (синхронное и (или) асинхронное) осуществляется через личный кабинет студента в ЭИОС, электронную почту.

7.2 Современные профессиональные базы данных:

Не используются.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

8.1 Перечень программного обеспечения

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows и MS Office, и среда разработки Microsoft Visual Studio.

Использование специализированного программного обеспечения для изучения дисциплины не требуется.

8.2 Информационные справочные системы

Не используются.

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для реализации дисциплины используются специальные помещения:

1. Учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, промежуточной и итоговой аттестации;
2. Помещения для самостоятельной работы обучающихся;
3. Помещения лабораторного практикума, оснащенного оборудованием необходимым для проведения экспериментальной части

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду НГУ.

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине

10.1 Порядок проведения текущего контроля и промежуточной аттестации по дисциплине

Текущий контроль успеваемости:

Текущий контроль осуществляется в ходе семестра путем опроса в начале каждой лекции по материалам предыдущей лекции, а также проведения опроса студентов в начале каждого занятия на темы, рассмотренные на предыдущем занятии.

Промежуточная аттестация:

Окончательная оценка работы студента в течение семестра происходит в рамках промежуточной аттестации с учетом результатов текущего контроля успеваемости обучающегося в семестре.

Критерии и шкалы оценивания индикаторов достижения результатов обучения отражены в Таблице 10.2.

Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине «Основы программирования»

Таблица 10.1

Код компетенции	Индикатор	Результат обучения по дисциплине	Оценочное средство
ОПК-3 (СУОС 03.03.02 Физика)	ОПК – 3.2. Применяет основные приемы, возможности и правила работы со стандартными и специализированными программными продуктами при решении	Знать: фундаментальные основы структурного подхода в программировании; основы работы с файловым вводом-выводом; принципы разработки консольных приложений; основы текстовой графики; принципы организации современных вычислительных систем.	Опрос Дифференцированный зачет

	профессиональных задач.	Уметь оперировать с основными типами данных и конструировать из них новые типы; работать с динамическими структурами данных: массивами и списками; оценивать сложность алгоритмов по времени и по памяти; ставить корректные технические задания.	
ОПК-4 (ФГОС 03.03.01 Прикладные математика и физика)	ОПК - 4.2. Применяет программно-технологические платформы, сервисы и информационные ресурсы информационной системы, в том числе, специализированных патентных баз данных для обработки научной и технической информации при решении фундаментальных и прикладных задач.	Владеть принципами структурного и объектно-базированного программирования; возможностью оценивать трудозатраты по программной реализации задачи.	

Таблица 10.2

Критерии оценивания результатов обучения	Шкала оценивания
<p>Ответ дан на все вопросы билета:</p> <ul style="list-style-type: none"> - самостоятельность, осмысленность, структурированность, логичность и аргументированность изложения материала, отсутствие затруднений в объяснении процессов и явлений, а также при формулировке собственных суждений, – точность и корректность применения терминов и понятий, – наличие исчерпывающих ответов на дополнительные вопросы. <p>При изложении ответа на вопрос(ы) преподавателя обучающийся мог допустить не принципиальные неточности.</p>	<i>Отлично</i>
<p>Ответ дан на все вопросы билета:</p> <ul style="list-style-type: none"> - самостоятельность, осмысленность, структурированность, логичность и аргументированность изложения материала, отсутствие затруднений в объяснении процессов и явлений, а также при формулировке собственных суждений, – точность и корректность применения терминов и понятий, – наличие ответов на дополнительные вопросы. <p>При изложении ответа на вопрос(ы) преподавателя обучающийся мог допустить неточности.</p>	<i>Хорошо</i>

<p>Ответ дан на вопросы билета:</p> <ul style="list-style-type: none"> - самостоятельность, осмысленность, структурированность, логичность и аргументированность изложения материала, отсутствие затруднений в объяснении процессов и явлений, а также при формулировке собственных суждений, – неточность при применении терминов и понятий, <p>При изложении ответа на вопрос(ы) преподавателя обучающийся допускает ошибки.</p>	<p><i>Удовлетворительно</i></p>
<p>Ответ дан не на все вопросы билета:</p> <ul style="list-style-type: none"> – фрагментарное и недостаточное представление теоретического и фактического материала, не подкрепленное ссылками на научную литературу и источники, – непонимание причинно-следственных связей, – отсутствие осмысленности, структурированности, логичности и аргументированности в изложении материала, – грубые ошибки в применении терминов и понятий, – отсутствие ответов на дополнительные вопросы. 	<p><i>Неудовлетворительно</i></p>

Типовые контрольные задания и иные материалы, необходимые для оценки результатов обучения

Задачи для решения

1. ***Программа решения квадратного уравнения (усвоение оператора ветвления и контроля типов).***
 - целочисленные коэффициенты задаются пользователем с клавиатуры
 - разветвление решения на случай линейного и квадратного уравнения
 - контроль корректности деления и извлечения квадратного корня
2. ***Печать всех простых чисел, не превышающих N (усвоение вложенных операторов цикла)***
 - N задается пользователем с клавиатуры
 - точная реализация математического определения: «имеет 2 делителя»
 - собственная реализация проверки делимости через приведение типов
 - методы оптимизации вычисления: 0 делителей, ограничение цикла.
3. ***Вычислить число π с заданной точностью (кол-во знаков после запятой), используя ряд Грегори***
 - вычисления суммы бесконечного ряда, не зная количества необходимых членов
 - собственная реализация правила округления
 - проверка критерия останова счета
 - вывод количества потребовавшихся членов ряда
4. ***Программа для решения уравнения вида $F(x) = 0$ методом Ньютона.***
 - Функция $F(x)$ задается преподавателем
 - Функция и ее производная задаются в тексте программы в виде отдельных функций

float Func(float x) и float dFunc(float x)
 - Начальное приближение и требуемая точность вычисления задаются с клавиатуры
 - Проверьте свой ответ, используя открытые ресурсы в Интернете

5. **Вычисление интеграла функции $F(x)$ методом трапеций**
 - Используется та же функция, что и задаче 4
 - Функция задается в тексте программы в виде отдельной функции float Func(float x)
 - шаг (количество разбиений) задается пользователем с консоли.
 - Оцените точность вычисления
 - Проверьте свой ответ, используя открытые ресурсы в Интернете
6. **Программа по вычислению максимума, минимума, среднего значения, среднеквадратичного отклонения во введенном статическом массиве**
 - максимальный размер массива задается #define, а размер – пользователем
 - хранение массива в виде глобальной переменной
 - целочисленные элементы массива задаются пользователем с клавиатуры
 - написание собственных функций, передача параметров, возвращение значений
 - используйте указатели и оператор * в функциях для расчёта минимума и максимума (Min, Max)
 - используйте массивы и оператор [] в функциях для расчёта среднего и среднекв. отклонения (Mean, RMS)
7. **Программа по вычислению максимума, минимума, среднего значения, среднеквадратичного отклонения во введенном динамическом массиве**
 - размер массива задается пользователем
 - целочисленные элементы массива задаются пользователем с клавиатуры
 - выделение и освобождение памяти в функции main ()
 - перепишите Min, Max из предыдущей задачи через [], а Mean, RMS – через *
8. **Сортировка введенного динамического массива**
 - реализация алгоритма сортировки пузырьком и его оптимизации
 - направление сортировки задает пользователь
 - реализация единой функции сортировки в соответствии с математической моделью, за счет выделения функций Compare () и Replace ()
 - реализуйте в программе несколько функций сравнения, для различных алгоритмов сортировки (по убыванию, по возрастанию, по абсолютному значению), и передайте указатель на соответствующую функцию сравнения в единую функцию сортировки
9. **Работа с матрицами 3x3**
 - Структура Matrix_3x3 для хранения данных
 - Данные хранятся в виде статического двумерного массива (float data [3][3])
 - Функции вычисления суммы, произведения, определителя и обратной матрицы
 - Исходная матрица заполняется целыми числами
 - Добавьте возможность чтения матрицы из файла (название задает пользователь с экрана; можно использовать функцию из задачи 10)
 - Результат операций выводится на экран и сохраняется в файл
10. **Работа с матрицами NxM**
 - Структура Matrix_NxM для хранения данных
 - Данные хранятся в виде динамического массива.
 - Функция InitMatrix задает размер и выделяет память.
 - Функция PrintMatrix выводит матрицу на экран в табличном виде
 - Функции вычисления суммы, произведения, определителя и обратной матрицы
 - Исходная матрица должна заполняться из файла или случайными числами
 - Результат операций выводится на экран и сохраняется в файл

- Вычисление определителя с использованием рекурсивной функции (разложение по строке)

11. *Обработка текста, введенного пользователем*

- текст хранится в статическом массиве заданного размера (ограничение на размер вводимого текста)
- реализация функции коррекции текста:
 - 1) большая буква только в начале предложения или в начале текста,
 - 2) удаление нескольких пробелов, следующих подряд,
 - 3) удаление пробелов перед точкой и запятой,
 - 4) добавление пробела после знаков препинания.

12. *Реализация функции ввода текста произвольного размера: char* GetText(FILE* stream, char terminator)*

- Конец текста обозначается выбранным символом terminator (например, '#' или '\n')
- Возвращается строка в стиле C (нуль-терминированная), не включающая символа terminator
- Выделение точного количества необходимой памяти (собственная функция перевыделения памяти без использования realloc())
- Использование функции корректировки из задачи 9
- Ввод текста из консоли или файла (задается с консоли)

13. *Записная книжка*

- Список из структур данных CPeople
- Поля данных: имя, фамилия, номер телефона, дата рождения
- Функции добавления элемента, удаления (по имени-фамилии), сортировки (по любому полю)
- Собственная реализация функции сравнения строк
- Реализация «пользовательского меню» (действие выбирает пользователь с консоли)
- Функция сохранения списка в файл и загрузки из файла (с добавлением либо заменой по выбору пользователя)

14. *Реализация короткого числа с плавающей точкой*

- Написать упаковщик и распаковщик чисел float во float8
- float8 содержит один бит знака – старший бит, несколько битов экспоненты (exponent), несколько битов мантиссы (significand or mantissa) – младшие биты.
- Количества бит для мантиссы и экспоненты, а также вычитаемое для степени (exponent bias) задаются константами препроцессора, которые можно настраивать.
- Требуется охватить все 256 значений – нормализованные (normal) и денормализованные (denormal or subnormal) числа, два нуля, две бесконечности, NaN-ы.
- NaN-ы при преобразованиях разрешается не различать и оставить один.
- При преобразовании числа, которое не выражается точно через float8, кодировать с минимальной потерей точности.
- Основные черты интерфейса:

```
#define EXP_SIZE
#define MANT_SIZE
#define EXP_BIAS
typedef int float8; /* 0..255 */
float F8toF32(float8);
float8 F32toF8(float );
```

15. Игра «собери яблоки» в случайном лабиринте

- Размер поля соответствует размеру консоли
- Края поля выделены символами «|» и “-”
- Пользователь задает процент заполнения препятствиями
- Корректность заполнения при 1%, 99% и 100%
- Карта препятствий хранится в отдельной структуре Field (дин. массив данных)
- Контроль невозможности наступить на препятствия и выйти за границу
- 2 типа препятствий: съедобные яблоки и стена (процент задает пользователь)
- Наличие входа и выхода из лабиринта – контроль проходимости
- Сохранение/восстановление карты и состояния игры

Вопросы к зачету

- Основы структурного программирования. Подход к решению задач
- Функции и массивы данных. Динамические массивы. Указатели
- Динамические массивы. Указатели
- Указатели и ссылки. Алгоритм сортировки пузырьком
- Системы счисления данных.
- Пользовательские типы данных
- Списки структур данных

Работа с файлами. Контроль динамической памяти

Оценочные материалы по промежуточной аттестации предназначенные для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС, хранятся на кафедре-разработчике РПД в электронном виде.